Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

TikogAI: A Feature-Engineered CNN Model for Classifying Indigenous Tikog Leaves in Banig Weaving

Las Johansen B. Caluza[1], Arnel C. Fajardo[2]

College of CSICT, Isabela State University – Cauayan Campus, Cauayan City, Philippines.
E-mail: [1]lasjohansencaluza@lnu.edu.ph, [2]acfajardo@gmail.com

Abstract

Innovative agricultural technologies increasingly utilize artificial intelligence (AI) and machine learning to enhance productivity and precision. Among these advancements, Convolutional Neural Networks (CNNs) have demonstrated significant promise in image classification tasks across various domains, including agriculture. However, the classification of Tikog leaves a culturally significant raw material used in the banig weaving industry in the Philippines has not been explored using CNNs with feature engineering. This study developed and optimized a feature-engineered CNN model for Tikog leaf classification by integrating Lab color space representation, data augmentation, autoencoder-based feature extraction, mean-max pooling, and dropout regularization. A total sample size of 500 standard-quality and 500 substandard-quality Tikog leaf images was augmented to generate 3,000 training images and 500 validation samples. Among the 27 CNN configurations tested, four models demonstrated superior performance, with Case 12 emerging as the best. This model achieved training and validation accuracies of 94.23% and 96.83%, F1-scores of 94.35% and 96.87%, ROC/AUC scores of 98.18% and 99.40%, and low sum of squared errors (SSE) values (173, 19). Case 12 exhibited excellent generalizability, high classification performance, and computational efficiency, making it the most effective model for deployment in real-world Tikog quality assessment. The study advances both technological innovation and the preservation of indigenous knowledge through intelligent systems.

DOI: https://doi.org/10.36548/jiip.2025.2.006
Received: 15.04.2025, received in revised form: 12.05.2025, accepted: 27.05.2025, published: 06.06.2025

Keywords: Autoencoder-based Feature Extraction, Leverage Artificial Intelligence, Image Processing, Deep Learning, Neural Network, Classification, Color Representation, Mean-Max Pooling, Dropout, CNN, Cultural Preservation.

1. Introduction

Convolutional Neural Networks (CNNs) have shown promise in image classification tasks, outperforming traditional methods that rely on feature engineering (Yadav, Madur, Dongare, Rajopadhye, & Salatogi, 2022). However, CNNs can exhibit cognitive gaps, leading to irrational behavior in visual recognition applications (Vietz, Rauch, Löcklin, Jazdi, & Weyrich, 2021). To address this, researchers have proposed methodologies to identify and close these gaps. One approach involves creating worst-case images using augmentation techniques to reveal potential cognitive weaknesses (Vietz, Rauch, Löcklin, Jazdi, & Weyrich, 2021). Another method utilizes synthetic training data generation through 3D rendering to target specific use-cases and improve CNN performance (Vietz, Rauch, Löcklin, Jazdi, & Weyrich, 2021). To enhance classification accuracy, novel CNN architectures have been developed, incorporating parallel convolutional layers with different filter sizes and global average pooling to reduce overfitting (Al-Sabaawi, Ibrahim, Arkah, Al-Amidie, & Alzubaidi, 2020). These advancements contribute to more robust and reliable image classification systems, particularly in challenging domains such as autonomous driving and manufacturing quality control.

In contrast, a study conducted by Milan (2021) proposed a deep learning-based image classification technique with 360 datasets consisting of 3334 images. The study applied feature detectors to the input image to generate the feature maps using the activation function. Although a promising outcome was produced, the researcher suggested that the model performance could be optimized by employing many training and testing images supported by a study that utilized a Convolutional Neural Network (CNN) for image classification and noted some challenges; therefore, a new encoding scheme for variable length was proposed such as a new representation of weight initialization strategy, and an adequate fitness evaluation method to speed up evolution (Bharadiya, 2023). The same study suggests an evolutionary algorithm that will address the classification feature extraction challenge and reduce the number of parameters required for this operation (Bharadiya, 2023). In addition, another study introduced CNN classification methods such as MobileNetV2, VGG16, InceptionV3, and TL-

Mobilev2, showing a significant outcome regarding 40 types of fruits for classification; however, feature reduction was not applied, which could cause overfitting (Gulzar, 2023). Despite strong performance, these pretrained models often lack domain-specific feature tuning, and without feature reduction steps like autoencoders, they risk overfitting due to high model complexity. Lastly, there are long-overdue issues related to overfitting, computational efficiency, and complexity in Convolutional Neural Network (CNN) models, including pre-trained models.

Lastly, ongoing research focuses on improving CNN architectures and exploring their applications across various fields, including agriculture and medical diagnosis (Sornam, Muthusubash, & Vanitha, 2017). In the agricultural domain, specifically for classifying Tikog leaves, a raw material used in the banig weaving industry, utilizing feature engineering and optimization of Convolutional Neural Network (CNN) remains unexplored. A banig is a traditional handwoven mat commonly used for sleeping and sitting in Asian countries, including the Philippines. In the Philippines, banig weaving is particularly renowned in Basey, Samar (Banig, 2013), utilizing tikog leaves. These leaves, sourced from a special reed grass that thrives in the swampy areas along rice fields, are dyed in intense, vivid colors before being woven into mats. While CNN-based classification has shown promise in agriculture, there remains a gap in integrating deep learning with indigenous material processing, particularly in the context of cultural industries. The lack of intelligent systems in indigenous industries like banig weaving is evident, and the gap lies in the absence of CNN-based classification models for Tikog leaves, Thus, this study addresses this gap by applying a feature engineered CNN model to classify Tikog leaves, the primary raw material used by the Basiao Native Weavers Association (BANWA) in the production of traditional banig. Moreover, the research addresses the unexplored application of a feature-engineered CNN model that combines Lab color space, autoencoder-based feature extraction, mean-max pooling, and dropout regularization for classifying Tikog leavesa culturally significant indigenous materialinto standard and substandard quality. To our knowledge, this integrated approach is novel and has not been previously applied to indigenous agricultural materials.

1.1 Objective of the Study

The primary objective of this study is to develop feature-engineered convolutional neural network (CNN) model for classifying Tikog leaves—an indigenous raw material used

in banig weaving—into standard and substandard quality categories. Specifically, this study seeks to:

- Develop a CNN model for classifying Tikog leaves into standard and substandard quality.
- Utilize Lab color space for enhanced color-based image representation.
- Apply autoencoder-based feature extraction for dimensionality reduction.
- Implement mean-max pooling to improve feature representation.
- Integrate dropout regularization for reducing overfitting and improving generalization.
- Evaluate the model's performance using accuracy, F1-score, ROC/AUC, and SSE.

2. Related Work

Studies explore CNN optimization and cognitive gaps in classification tasks. Studies have identified cognition gaps in visual recognition applications using worst-case image generation techniques CNN Optimization and cognitive Gaps in Classification (Vietz, Rauch, Löcklin, Jazdi, & Weyrich, 2021). However, these approaches focus primarily on identifying weaknesses without providing integrated solutions for practical deployment in niche applications such as cultural material classification. Various optimization strategies, including genetic algorithms and particle swarm optimization, have been employed to enhance CNN performance (Chatterjee, Akhtar, & Pradhan, 2021; Sharma & Kumar, 2023). Hybrid models combining CNNs with genetic algorithms have shown improved accuracy in brain tumor classification (Özdem, et al., 2022). Bio-inspired optimization techniques have demonstrated success in improving CNN performance for breast cancer detection in infrared images reported VGG-16 F1-Score value of 92% and ResNet-50 F1-Score value of 90% (Gonçalves, Souza, & Fernandes, 2022). While bio-inspired optimization improved classification metrics, the study did not provide insights into memory consumption or computational overhead (e.g., FLOPs), which limits scalability. Moreover, hyperparameter optimization improved CNN classification accuracy by 6% (Wojciuk, Swiderska-Chadaj, Siwek, & Gertych, 2022). Continuous CNN optimization has proven significant improvement specifically on hyperparameter tuning.

Feature engineering techniques play a crucial role in enhancing the performance of Convolutional Neural Networks (CNNs) across various applications. Traditional approaches involve hand-crafted feature extraction methods, which require human expertise but can yield good results in tasks like plant disease classification (Gunarathna & Gunarathna, 2020). However, deep learning methods, particularly CNNs, have emerged as powerful automatic feature extractors, achieving high classification rates without manual intervention (Arunachalam & Karthikayani, 2020). While these methods show improved accuracy, many do not address computational efficiency or fail to incorporate pooling and regularization techniques that reduce overfittingfactors that are essential for real-world applications in constrained environments. Moreover, studies have explored various techniques such as Discrete Wavelet Transform for DeepFake detection (Byrroughs, Gokaraju, Roy, & Khoa, 2020) and efficient indexing methods for high-dimensional CNN features (Saini, Gupta, & Kaur, 2023). The integration of feature selection with CNN architectures has shown promise in breast cancer analysis (Gupta, 2024), while the importance of feature engineering in improving machine learning solutions is emphasized across domains (Duboue, 2020). Existing works primarily emphasize storage and retrieval improvements and often neglect the role of feature reduction in improving classification accuracy and computational efficiency during training. Therefore, feature-engineering concepts in image classification remain an interesting area in deep learning because of their wide array of probabilities to contribute to improving image classification algorithms.

Techniques to address overfitting in neural networks, remain a critical challenge in machine learning. Dropout, a method that randomly removes units during training, has emerged as an effective regularization technique (Shirke, Walika, & Tambade, 2018; Lim, 2021). It prevents co-adaptation of units and improves generalization across various tasks (Noh, You, Mun, & Han, 2017; NarasingaRao, Venkatesh Prasad, Sai Teja, Zindavali, & Phanindra Reddy, 2018). Moreover, it is reported that dropout can mitigate both underfitting and overfitting in neural networks by applying it at the start or end of training, respectively (Liu, Xu, Jin, Shen, & Darrell, 2023). However, Liu et al. (Liu, Xu, Jin, Shen, & Darrell, 2023) suggested that further research in developing neural network regularizers specifically on deeper understanding of early dropout's effect on training loss, optimization, and factors like training duration or optimizer choice. Most dropout studies focus on general-purpose datasets and do not explore

dropout's interaction with handcrafted or hybrid features in indigenous material contexts, which this study uniquely addresses.

3. Proposed Work

Deep learning concepts in image processing focus on image extraction, embedding, and creating layers for neural networks utilizing two significant frameworks: the sequential API and the functional API (Manaswi, 2018). The process was adopted to perform an image classification problem, as shown in Figure 1:

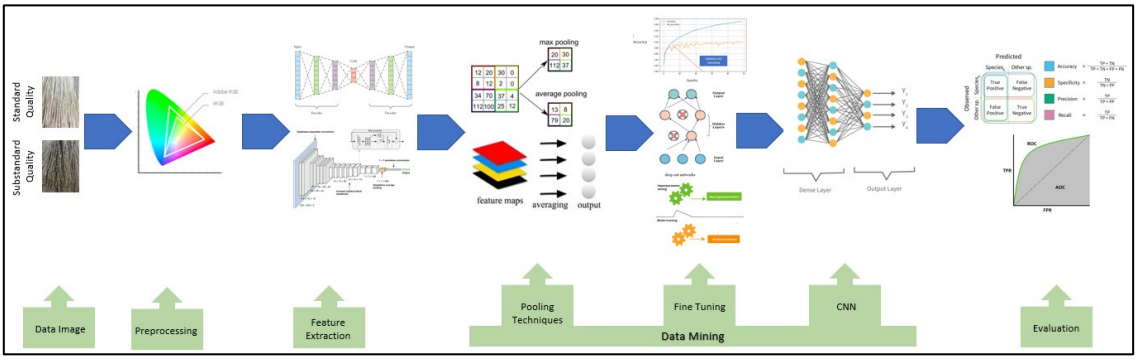

Figure 1. Tikog Leaf Classification Through Knowledge Discovery in Databases

3.1 Data Collection and Selection

The primary data utilized in the study was taken from the research locale, Basiao Native Weavers Association (BANWA), Basey, Samar, Philippines. The proponent requested the association's president to gather images of the tikog leaves. Tikog leaf is a wild grass used as a raw material in making a *banig,* also known as a mat, bag, wallet, and other similar handmade products by *lara* or weaving. The tikog leaves provided by the association were segregated. It was explained that the tikog categorized as*hinog,* also known as Standard Quality, shown in Figure 2, based on their assessment, while *hilaw,* also known as Substandard Quality, shown in Figure 3 refers to leaves that arefresh or have discoloration. Discoloration occurs when the pagbulad or leaf drying process is exposed to water or moisture. Thus, the quality of the tikog leaf, such as color and brittleness, is affected. Furthermore, the images were captured at different heights or distances, such as 1 foot, 2 feet, and 3 feet, and under different lighting conditions. The data collected from the research locale comprises 500 images for Standard Quality and 500 for Substandard Quality. The BANWA weavers of Basey, Samar, Philippines,

carefully identified and suggested these images. The collected images were stored in a Standard Quality and Substandard Quality folders.

Figure 2. Standard Quality of Tikog Leaf

Figure 3. Substandard Quality of Tikog Leaf

3.2 Preprocessing

This study utilized the CIELAB color space for image enhancement, ultimately impacting image segmentation performance. Regarding its performance, results show that the CIELAB space method has better accuracy than the HSV method (Setyawan, Riwinanto, Nursyahid, & Nugroho, 2018). The HSV method has an average accuracy of 75.33%, while the CIELAB space method has a greater average accuracy of 78.39% (Setyawan, Riwinanto, Nursyahid, & Nugroho, 2018). Furthermore, results revealed that the L*a*b* color space can identify good or bad patterns of images and compare its performance with the HSV space that the authors recently reported (Prasad, 2017). It is shown that the L*a*b* space outperforms the HSV space (Prasad, 2017).

3.2.1 Color Lab Space

Parameters:

ℓ - image,

H – height,

W – width;

Given an input image $\ell \in \mathbb{R}^{H\ X\ W\ X\ 3}$ in RGB color space, and three refers to the RGB channels. To convert the ℓ from RGB to Lab color space:

$$\ell_{Lab} = Lab(\ell) \tag{1}$$

Where $\ell_{Lab} \epsilon \mathbb{R}^{H\ X\ W\ X\ 3}$ and consists of three channels $L, a,$ and b.Thus, RGB to Lab Color Space transformation can be expressed as:

$$\ell_{Lab} = RGBtoLab(\ell) \tag{2}$$

Where ℓ_{Lab} is the image in the Lab color space. RGBtoLAB is a function that converts an image from RGB to Lab color space. The Lab color space separates the luminance (L) from the chrominance (a,b), which helps better feature extraction of color variations.

3.2.2 Augmentation

Data augmentation has become crucial for improving machine learning model performance and generalization. Recent research has focused on automated data augmentation methods, which outperform classical approaches (Mumuni & Mumuni, 2024). The ImageDataGenerator is highly optimized and leverages GPU acceleration for efficient preprocessing. The ImageDataGenerator in Keras is built on a series of image preprocessing and augmentation operations, each of which can be represented mathematically.

$$I' = T(I) = T_n \circ T_{n-1} \circ \ldots \circ T_1(I) \tag{3}$$

Where:

I is the input image

I' is the transformed output image

T is the composite transformation function composed of $T_1, T_2, \ldots T_n$, where each T_i represent an individual transformation (rotation, scaling, translation, etc.).

$\circ$ denotes function composition, meaning transformations are applied sequentially.

Thus, using this algorithm, the augmented data generated was 3000, 1500 for Standard images, 1500 for Substandard images, and 500 for testing/validation datasets.

3.3 Image Transformation

3.3.1 Image Size

In this study, the images were transformed into 224x224 pixels. In recent years, CNN have thrived and made significant breakthroughs in computer vision (Sangineto, Nabi, Culibrk, & Sebe, 2018; Tian, Li, Qu, & Yan, 2017; Qin, Pan, Xiang, Tan, & Hou, 2020). Current CNNs require a fixed input image size, such as 224 × 224 or 299 × 299 (Qin, Pan, Xiang, Tan, & Hou, 2020). In support of this, a study using the DenseNet-121 model achieves the highest accuracy in classifying chest X-ray images for various lung diseases using 224x224 pixel resolution (Rochmawanti & Utaminingrum, 2021).Additionally, cropping dermoscopic images to 224x224 and using a multi-scale, multi-network feature-engineered approach improves skin lesion classification performance (Mahbod, et al., 2020). Therefore, this study shall utilize the 224x224 image size.

Parameters:

For each pixel in the resized image Δ_{target}, the corresponding pixel intensity in the original image Δ_{orig} is given as:

$$\Delta_{target}(a_{target}, b_{target}) = \Delta_{orig}(\frac{a_{target}}{S_{width}}, \frac{b_{target}}{S_{height}}) \quad (4)$$

The intensity of a pixel at position (a_{target}, b_{target}) in the resized image Δ_{target} is obtained by mapping back to the corresponding coordinates in the original image Δ_{orig}, scaled by the width and height factors.

3.3.2 Feature Extraction Using Autoencoders

Let ℓ_{Lab} the feature representation where each pixel has three components, such as L, a, b. The Autoencoders learn a compressed latent representation Z of the input data. For the Encoder, the algorithm is:

$$Z = f_\theta(\ell_{Lab}) = \sigma(W_e \ell_{Lab} + b_e) \quad (5)$$

Where:

ℓ_{Lab} is the input Lab color image.

W_e, b_e are the weights and biases of the Encoder.

σ is the activation function (commonly Sigmoid or ReLU).

Z is the compressed latent representation (features).

Once trained, we only use the Encoder f_θ to obtain the latent feature vector Z, which is a compressed version of the input (Patel & Upla, 2019; Berahmand, Daneshfar, Salehi, Li, & Xu, 2024; Chen & Guo, 2023)

3.3.3 Autoencoders Pooling Technique Using Mean-Max

Autoencoders have been explored with various modifications to enhance feature extraction and representation learning. Pooling layers reduce spatial dimensions, computational costs, and overfitting (Gholamalinezhad & Khosravi, 2020; Zafar, et al., 2022). One of the pooling techniques is the Max pooling. Max pooling is a mechanism that optimizes the spatial size of a feature map while also providing the network with translation invariance (Zafar, et al., 2022). The problem is that max pooling only considers the largest element and ignores the others, as we can see in the example (Zafar, et al., 2022). In some cases, after max pooling, salient features disappear when most elements have high magnitudes, which can lead to unacceptable results (Zafar, et al., 2022; Li, Yang, feng, Chakradhar, & Zhou, 2016). Another technique is average pooling, where the input is segmented into rectangular pooling areas, and an average pooling layer downsamples by calculating the average values of each region (Lecun, Bottou, Bengio, Haffner, & Patrick, 1998). The problem is that it declines in information in terms of contrast where all activation values in the rectangular box are considered when estimating the mean. The estimated mean will indeed be low if the strength of all the activation functions is low, resulting in diminished contrast. The problem worsens once most of the activations in the pooling zone have a zero value (Zhang, Li, Peng, Chen, & Zhang, 2018). Mean-max attention autoencoders utilize both mean and max pooling operations to capture diverse information (Zhang, Wu, Li, & Li, 2018). Convolutional autoencoders with max-pooling layers perform better in digit and object recognition tasks (Masci, Meier, Dan, & Schmidhuber, 2011) compared to other pooling techniques. Thus, this study shall utilize each patch's mean-max pooling of autoencoders with $Z_{max} = max_{i,j}\left(Z_p^{(i,j)}\right)$ and $Z_{mean} = \frac{1}{p^2}\sum_{i,j} Z_p^{(i,j)}$. Where Z_{max} is the maximum value over the patch; while the Z_{mean} is the average of all the values in the patch. Then combine with the weighted sum as $Z_{mean-max} = \alpha Z_{max} + (1-\alpha)Z_{mean}$. The final pooled output is the latent space representation:

$$Z_{latent} = Z_{pooled}^{(L)} \tag{6}$$

3.3.4 Dropout

Dropout fine-tuning in image classification has shown promising results, such as autotuning hyperparameters, including dropout rates, which can improve performance of self-supervised models (Kishore & Mukherjee, 2024). Furthermore, high dropout rates during fine-tuning can enhance distribution performance (Zhang & Bottou, 2024). Dropout is applied to prevent overfitting by randomly deactivating a fraction of neurons during training (Gulzar, 2024; Peng, et al., 2024). The F1-score, reflecting a balance between precision and recall, exhibits remarkable performance across all classes, with values ranging from 0.9851 to 1.0000. (Gulzar, 2023). Thus, this study shall utilize the Dropout Regularization after extracting the feature map Z_{conv1}. Thus, let p be the dropout rate and the dropout operation shall be

$$Z_{dropout} = Z_{conv1}\, x\, M \tag{7}$$

Where $M \sim Bernoulli(1-p)$ is a binary mask with probability of p of setting units to zero.

3.4 Data Mining

To illustrate the foundation of the neural network in this study, Figure 4 shows the input layer, hidden layer, bias, activation, and output following the equation where the input is multiplied by the respective weights and added, and bias is added to the result. Then, an activation function, g, is applied so that the output of the neuron is g(w·x+b) (Chen, Duan, Kang, & Qiu, 2021; Tian, Li, Qu, & Yan, 2017):

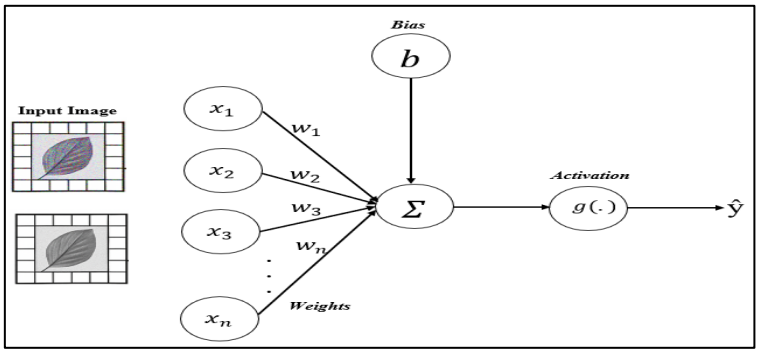

Figure 4. Neural Network Diagram

$$x = \sum_{i=0}^{n} w_i x_i + b = w_0 x_0 + w_1 w_1 + \ldots + w_n b_n + b = w.x + b \tag{8}$$

3.5 Evaluation

In this stage, the detected patterns shall be revealed, whether they are interesting or not. This study utilized the confusion matrix that shows the precision (*specificity*) and recall (*sensitivity*), the area under the curve (AUC), and finally, the model performance before and after the application of the factorial analysis algorithm.

Accuracy is the fraction of the number of correctly predicted examples to the total number of instances in the dataset. It is given by:

$$Accuracy = \frac{Number\ of\ correct\ predictions\ (TP+TN)}{Total\ number\ of\ predictions\ (TP+TN+FP+FN} \quad (9)$$

Precision is the ratio of true positive, relevant instances to the total number of retrieved instances. It is given by:

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

Recall is also called sensitivity, which is the fraction of correct positive examples predicted out of the total number of positive occurrences.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

The F1-score, also known as the F-score or F-measure, is a metric used to evaluate a machine learning model's performance. It combines precision and recall into a single score.

$$F - score = 2 * \frac{precision*recall}{precision+recall} \quad (12)$$

4. Results and Discussion

4.1 Tikog Leaf Classification Model Performance

Based on the simulation results to test the optimized CNN model using various algorithms such as Color Lab Space for color representation, Autoencoders for feature extraction, Mean-Max Pooling for the pooling technique, and Dropout for fine-tuning four (4) models emerged from 27 models and cases utilizing various scenarios such as the number of epochs ranging from 100 to 200, learning rate ranging from 0.01 to 0.0001, and CNN activation

functions such as sigmoid, hyperbolic tangent (tanh), and relu as shown in Table 1. The four (4) cases that had better performance than the rest were case 5, case 11, case 12, and case 15.

For Case 5, it shows that a 150-epoch training with a learning rate pegged at 0.001 using the sigmoid function as its CNN activation has an augmentation time of 26.85 sec, training time of 1106.04 sec, evaluation time of 9.81 sec, and validation time of 5.72 sec resulting in a training performance of 96.97% accuracy, 96.62% precision, 97.33% recall, 96.98% F1-score, and 99.59% ROC/AUC with a Sum of Squares Errors (SSE) of 91, demonstrating significant performance as an image classification model. However, when testing its model, the results show 90% accuracy, 85.21% precision, 96.80% recall, 90.6% F1-score, and 98.32% ROC/AUC with a Sum of Squares Error (SSE) of 50. These results imply that while Case 5 demonstrates strong performance in the image classification model based on its training results, it struggles with precision in the test data. This indicates that the model may slightly overfit the training data, as it is more prone to false positives when exposed to unseen data. Further fine-tuning in the activation function, learning rate, and regularization can be applied to improve its generalizability.

As illustrated in Table 1, Case 11 shows that with 100 epochs, a learning rate of 0.001, and using hyperbolic tangent as its activation function, the model achieved a 28.31 sec augmentation time, 723.03 sec training time, 9.08 sec evaluation time, and 6.09 sec validation time, resulting in 98.03% accuracy, 96.45% precision, 99.73% recall, 98.07 F1-score, and 99.89% ROC/AUC with a Sum of Squares Errors (SSE) of 59. However, the testing set declined to 89.80% accuracy, 93.07% precision, 86% recall, 89.40% F1-score, and 96.27% ROC/AUC with a Sum of Squares Errors (SSE) of 51. These results imply strong performance in the training model, with exceptional accuracy, recall, and F1-score. However, the decline in testing performance highlights challenges in the model's generalizability, particularly regarding accuracy rate and precision. The significant decrease in recall (99.73% in training vs. 86% in testing) suggests that the model struggles with identifying Standard Tikog leaves in unseen data. This is a key indicator of overfitting, where the model has learned specific patterns from the training data that do not generalize well to new samples. Thus, the results indicate the likelihood of overfitting caused by the higher learning rate, tanh activation, and epoch count. Further tuning can be achieved by utilizing hyperparameter tuning or lowering the learning rate and increasing the epoch count.

Results shown in Table 1 indicate that Case 12, with 100 epochs, a learning rate of 0.0001, and hyperbolic function (tanh) as its activation function, resulted in 25.04 sec augmentation time, 756.89 sec training time, 8.71 sec evaluation time, and 7.07 sec validation time, generating 94.23% accuracy, 92.45% precision, 96.33% recall, 94.35% F1-score, and 98.18% ROC/AUC with a Sum of Squares Errors (SSE) of 173 during the training of the optimized CNN model. Results from the validation of the optimized model show an increase in all validation performance metrics, such as 96.83% accuracy, 95.77% precision, 98% recall, 96.87 F1-score, and 99.40% ROC/AUC with a Sum of Squares Errors (SSE) of 19. Case 12 results imply an exceptional model showing a highly effective, optimized CNN model based on the parameters. The low training and validation time suggest an efficient model despite its complexity and low reported error; thus, there is computational efficiency and stability during the training and testing of the model, and it quickly adapts to unseen data while maintaining stable performance. Furthermore, the optimized CNN model suggests excellent precision and recall based on the F1-score and Receiver Operating Characteristic/Area Under Curve (ROC/AUC) metrics in identifying, determining, and handling false positives and false negatives during training and validation data. Finally, the results imply that the model's generalizability was exceptional, even better on the unseen data, which contained fewer ambiguities or noise in both datasets.

In the last result, Case 15 was simulated with the following parameters: 150 epochs, a learning rate of 0.0001, and hyperbolic (tanh) as its activation function. In this context, the generated augmentation time is 24.09 seconds, 1162.87 seconds training time, 8.93 seconds evaluation time, and 7.16 seconds validation time. As a result, it shows 95.52% accuracy, 92.94% precision, 98.52% recall, 95.65% F1-score, and 99.18% ROC/AUC with a Sum of Squares Errors (SSE) of 153 during the training of the optimized CNN model. Meanwhile, the results during validation were 93.40% accuracy, 88.34% precision, 100% recall, 93.81% F1-score, and 99.78% ROC/AUC with a Sum of Squares Errors (SSE) of 23. The results imply that the optimized CNN model demonstrates a well-optimized approach for tikog leaf classification, given that the simulation revealed efficient time performance in all parameters relative to time performance. The training results show a promising outcome in terms of accuracy, precision, recall, F1-score, ROC/AUC, and SSE, indicating minimal error. However, when validated, although it shows promising results in accuracy, recall, F1-score, ROC/AUC, and a significant drop in SSE, it lacks or has significantly reduced precision during validation,

indicating effects in identifying positive and false positive classes. Thus, the slightly reduced precision (88.34%) in validation implies a minor increase in false positives, which may require further refinement depending on the application's tolerance for misclassification.

Table 1. Model Performance Matrix of Tikog Leaf Classification

Cases	No. of Epoch	Learning Rate	CNN Activation	Time Performance (Seconds)				Training Data Metrics						Validation Data Metrics						Convergence Metrics		
				Augmen-tation	Training	Evalua-tion	Valida-tion	Accuracy	Precision	Recall	F1-Score	ROC /AUC	SSE	Accuracy	Precision	Recall	F1-Score	ROC /AUC	SSE	Epoch of Conver-gence	Converged Valida-tion Loss	Learning Rate at Conver-gence
1	100	0.01	Sigmoid	23.79	746.62	8.86	6.65	0.5000	0.0000	0.0000	0.0000	0.5000	1500.00	0.5000	0.0000	0.0000	0.0000	0.5000	300.00	7	0.6941	0.01
2	100	0.001	Sigmoid	27.13	772.75	9.33	7.05	0.9220	0.8667	0.9973	0.9275	0.9846	234.00	0.7200	0.6410	1.0000	0.7812	0.7512	168.00	1	0.6437	0.001
3	100	0.0001	Sigmoid	23.25	751.86	8.81	6.62	0.8277	0.7548	0.9707	0.8492	0.9511	517.00	0.4833	0.4864	0.5967	0.5359	0.5628	310.00	90	0.7017	0.0001
4	150	0.01	Sigmoid	24.13	1122.6	8.75	6.82	0.5000	0.0000	0.0000	0.0000	0.5000	1500.00	0.5000	0.0000	0.0000	0.0000	0.5000	300.00	1	0.6949	0.01
5	150	0.001	Sigmoid	26.85	1106.04	9.81	5.72	0.9697	0.9662	0.9733	0.9698	0.9959	91.00	0.9000	0.8521	0.9680	0.9064	0.9832	50.00	51	0.7884	0.001
6	150	0.0001	Sigmoid	23.57	1138.05	2.81	6.84	0.9363	0.9253	0.9493	0.9372	0.9820	191.00	0.8217	0.7406	0.9900	0.8474	0.9615	107.00	47	0.6591	0.0001
7	200	0.01	Sigmoid	23.51	1509.12	8.79	7.19	0.5000	0.5000	1.0000	0.6667	0.5000	1500.00	0.5000	0.5000	1.0000	0.6667	0.5000	300.00	12	0.6937	0.01
8	200	0.001	Sigmoid	23.93	1517.46	8.73	7.2	0.9817	0.9938	0.9693	0.9814	0.9984	55.00	0.8750	0.9747	0.7700	0.8603	0.9658	75.00	1	0.8873	0.001
9	200	0.0001	Sigmoid	27.71	1470.39	9.02	6.01	0.8617	0.8266	0.9153	0.8687	0.9487	415.00	0.6680	0.6780	0.6400	0.6584	0.7243	166.00	1	0.7066	0.0001
10	100	0.01	tanh	32.29	755.41	8.71	6.73	0.5000	0.0000	0.0000	0.0000	0.8069	1500.00	0.5000	0.0000	0.0000	0.0000	0.3812	300.00	1	16.1181	0.01
11	100	0.001	tanh	28.31	723.03	9.08	6.09	0.9803	0.9645	0.9973	0.9807	0.9989	59.00	0.8980	0.9307	0.8600	0.8940	0.9627	51.00	55	0.8553	0.001
12	100	0.0001	tanh	25.04	756.89	8.71	7.07	0.9423	0.9245	0.9633	0.9435	0.9818	173.00	0.9683	0.9577	0.9800	0.9687	0.9940	19.00	77	0.5063	0.0001
13	150	0.01	tanh	23.16	1125.57	8.59	6.64	0.5000	0.0000	0.0000	0.0000	0.5000	1500.00	0.5000	0.0000	0.0000	0.0000	0.5000	300.00	1	16.1181	0.01
14	150	0.001	tanh	23.25	1112.98	8.75	6.66	0.9577	0.9288	0.9913	0.9590	0.9957	127.00	0.8233	0.8031	0.8567	0.8290	0.9142	106.00	1	0.8075	0.001
15	150	0.0001	tanh	24.09	1162.87	8.93	7.16	0.9552	0.9294	0.9852	0.9565	0.9918	153.00	0.9340	0.8834	1.0000	0.9381	0.9978	23.00	43	0.315	0.0001
16	200	0.01	tanh	24.51	1529.64	8.89	6.94	0.5000	0.5000	1.0000	0.6667	0.5000	1500.00	0.5000	0.5000	1.0000	0.6667	0.5000	300.00	11	0.6924	0.01
17	200	0.001	tanh	24.51	1561.6	8.94	6.8	0.9083	0.8576	0.9793	0.9144	0.9807	275.00	0.8083	0.7428	0.9433	0.8311	0.8760	115.00	1	0.7904	0.001
18	200	0.0001	tanh	27.49	1551.64	9.06	7.22	0.8857	0.8350	0.9613	0.8937	0.9598	343.00	0.5583	0.5485	0.6600	0.5991	0.5943	265.00	139	1.1841	0.0001
19	100	0.01	Relu	28.35	767.59	8.87	7.00	0.5000	0.0000	0.0000	0.0000	0.5000	1500.00	0.5000	0.0000	0.0000	0.0000	0.5000	300.00	1	0	0.01
20	100	0.001	Relu	26.72	777.34	9.07	6.67	0.6787	0.6128	0.9707	0.7513	0.8701	964.00	0.4617	0.4695	0.5900	0.5229	0.4887	323.00	1	0.9071	0.001
21	100	0.0001	Relu	27.41	769.59	20.83	7.27	0.7943	0.8809	0.6807	0.7680	0.9128	617.00	0.7667	0.7899	0.7267	0.7569	0.8668	140.00	1	0.8692	0.0001
22	150	0.01	Relu	29.00	1122.88	8.93	6.67	0.5000	0.5000	1.0000	0.6667	0.4186	1500.00	0.5000	0.5000	1.0000	0.6667	0.5336	300.00	1	0	0.01
23	150	0.001	Relu	27.14	1157.56	9.3	7.21	0.7640	0.7683	0.7621	0.8576	0.5567	708.00	0.5567	0.5746	0.4367	0.4962	0.5277	266.00	1	1.0184	0.001
24	150	0.0001	Relu	27.26	1177.17	9.1	6.9	0.8503	0.8238	0.8913	0.8562	0.9252	449.00	0.9033	0.8623	0.9600	0.9085	0.9809	56.00	140	0.4435	0.0001
25	200	0.01	Relu	36.43	1544.54	8.69	7.54	0.5000	0.0000	0.0000	0.0000	0.5000	1500.00	0.5000	0.0000	0.0000	0.0000	0.5000	300.00	1	16.1181	0.01
26	200	0.001	Relu	27.47	1509.32	8.79	6.73	0.8913	0.9071	0.8720	0.8892	0.9637	326.00	0.8283	0.9581	0.6867	0.8000	0.8644	103.00	1	0.7033	0.001
27	200	0.0001	Relu	27.46	1500.85	8.69	7.01	0.8980	0.8571	0.9553	0.9035	0.9598	306.00	0.5050	0.5053	0.4800	0.4923	0.4750	297.00	1	0.9069	0.0001

4.2 Best Performing Optimized Model based on Training and Testing Datasets

Based on the results presented in Table 2, Case 12 reflects a well-tuned and optimized CNN model that balances accuracy (94.23%, 96.83%), precision (92.45%, 95.77%), recall (96.33%, 98.00%), F1-score (94.35%, 96.87%), ROC/AUC (98.18%), and SSE (173, 19), which is computationally cost-effective with minimized misclassification, as shown in Figures 5, 6, 7, and 8. Moreover, Model 12 maintains high classification reliability and excellent agreement beyond chance, with MCC and Cohen's Kappa values of 0.951, which strengthens its suitability for deployment in real-world applications. The Matthews Correlation Coefficient (MCC) value of 0.951 indicates a very strong correlation between the predicted and actual classifications of Tikog leaf quality. MCC is particularly useful for evaluating binary classification tasks on imbalanced datasets, as it considers all elements of the confusion matrix (true positives, true negatives, false positives, and false negatives). Furthermore, Cohen's Kappa value of 0.951 signifies almost perfect agreement between the predicted labels of the model and the actual labels, beyond what would be expected by chance. Kappa accounts for random agreement and is especially valuable in real-world classification tasks where class imbalance or subjective judgments might exist. Thus, reducing the risk of errors in real-world use, also known as generalization capability, makes it suitable for resource-constrained

environments or computational efficiency. Furthermore, the model suggests efficient learning without overtraining, which indicates that it has successfully learned from the data and captured the underlying patterns without requiring the entire 100 epochs and converging at the 77th epoch. The remaining 23 epochs likely served to fine-tune the parameters, ensuring stability and preventing deterioration due to overfitting. Thus, the optimization algorithms and parameters, or the chosen hyperparameters, were well-optimized. Therefore, the Case 12 optimized CNN model shows a non-overfitting nature with computational efficiency and is highly effective, reinforcing its suitability as the best-performing model among the cases discussed. Furthermore, the memory efficiency of CNNs has mainly been overlooked in previous work because the memory behavior of CNNs can have a significant impact on their performance (Li, Yang, feng, Chakradhar, & Zhou, 2016), including computational efficiency and memory (Rizvi, Rahman, Sheikh, Fuad, & Shehzad, 2023), and reduces computational complexity (Limonova, Sheshkus, & Nikolaev, 2016), where this study has answered this gap.

Moreover, based on the model configuration, the estimated computational complexity for Case 12 is approximately 12.62 GFLOPs (Giga Floating Point Operations). This value reflects the operations needed to process a single 224x224 Tikog leaf image through the network, as the convolutional and dense layers are complex. Relatively, it is a low computational cost for Case 12 and is interpreted as resource-efficient, making it suitable for deployment in environments with limited hardware capabilities. The model's ability to maintain high classification accuracy and generalization performance with minimal FLOPs supports its practicality for real-time quality control in indigenous industries, such as the banig weaving sector of Basey, Samar.

Table 2. Case 12 Performance Metrics

Performance	Training Data Metrics	Validation Data Metrics
Accuracy	94.23%	96.83%
Precision	92.45%	95.77%
Recall	96.33%	98.00%
F1-Score	94.35%	96.87%
ROC/AUC	98.18%	99.40%
SEE	173	19

Matthews Correlation Coefficient (MCC)		0.951
Cohen's Kappa		0.951

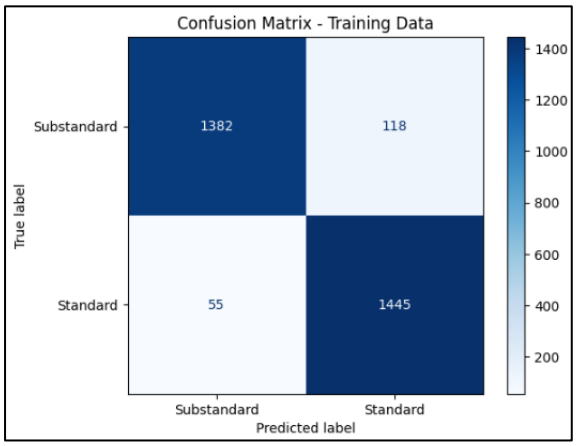

Figure 5. Case 12 Confusion Matrix in Training Data

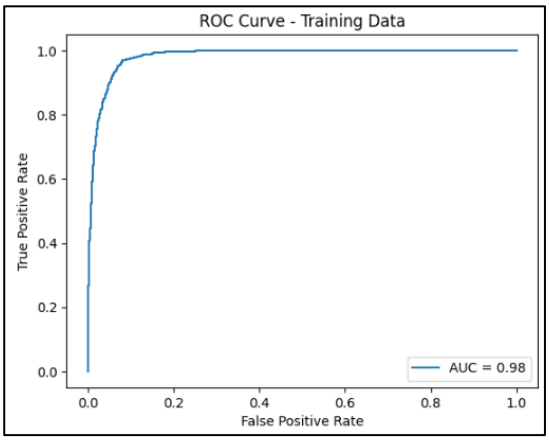

Figure 6. Case 12 ROC Curve in Training Data

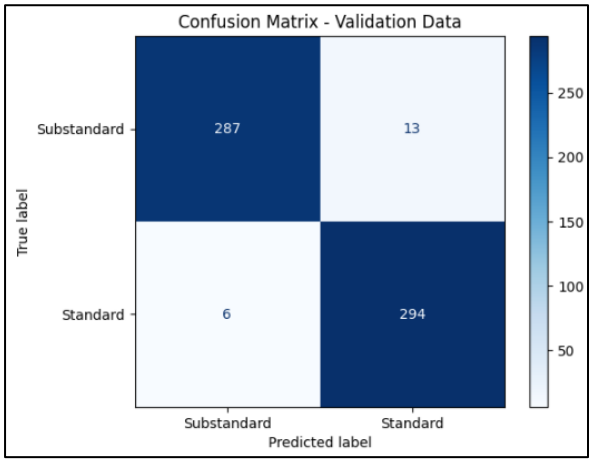

Figure 7. Case 12 Confusion Matrix in Validation Data

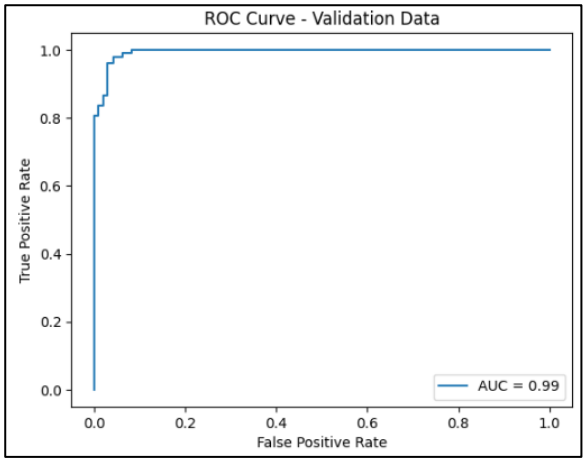

Figure 8. Case 12 ROC Curve in Validation Data

As shown in Table 3 and visualized in Figure 9, the training and validation loss curves of the four top-performing feature-engineered CNN models (Models 5, 11, 12, and 15) reveal distinct convergence behaviors and levels of generalization. Model 5 demonstrates a reasonable fit, with both training and validation losses decreasing over time; however, its relatively high converged validation loss of 0.7884 at epoch 51 suggests limited generalization and potential overfitting, as indicated by the divergence between training and validation curves. In contrast, Model 11 exhibits more pronounced overfitting, with a low training loss but a higher validation

loss of 0.8553 at epoch 55. This indicates a need for further regularization or learning rate adjustments. Model 15 achieves the most stable convergence and the lowest validation loss of 0.3150, occurring early at epoch 43. The close alignment of its training and validation losses suggests excellent model robustness and minimal overfitting, even over extended epochs. Ultimately, Model 12 offers the best generalization performance among all models, with a converged validation loss of 0.5063 at epoch 77. Its training and validation losses are well-balanced, reflecting a stable and well-regularized model. These findings suggest that while Model 15 excels in convergence speed and minimal error, Model 12 achieves the most reliable performance across varying data, making it the most suitable for real-world deployment in Tikog leaf quality classification. Among the tested components, the integration of Lab color space and autoencoder-based feature extraction provided foundational feature quality, enhancing the model's ability to distinguish color variations and structural patterns in Tikog leaves (Setyawan, Riwinanto, Nursyahid, & Nugroho, 2018; Prasad, 2017; Chen & Guo, 2023). Meanwhile, the use of mean-max pooling and dropout regularization ensured generalization and robustness by capturing diverse information and mitigating overfitting (Zhang, Wu, Li, & Li, 2018; Liu, Xu, Jin, Shen, & Darrell, 2023; Lim, 2021). These engineered features, when paired with carefully tuned hyperparameters such as learning rate, activation function, and number of epochs, notably improved the performance of Model 12 (Wojciuk, Swiderska-Chadaj, Siwek, & Gertych, 2022; Sharma & Kumar, 2023). This multi-layered enhancement is supported by contemporary studies in image classification, confirming that a hybrid of preprocessing, feature reduction, and regularization yields optimal deep learning outcomes in specialized domains such as agriculture and cultural heritage (Gulzar, 2023; Gonçalves, Souza, & Fernandes, 2022; Mahbod, et al., 2020).

The findings of this study have practical implications for both technological innovation and cultural preservation. The superior generalization ability of Model 12 suggests it is well-suited for real-world deployment in classifying Tikog leaf quality, which can enhance quality control processes for the banig weaving industry in Basey, Samar. This demonstrates how artificial intelligence can support indigenous industries by improving product consistency while preserving traditional craftsmanship (Gulzar, 2023; Chen & Guo, 2023). Furthermore, the study presents a model selection framework based on convergence behavior and validation loss, offering a replicable approach for similar image classification tasks in agriculture and cultural heritage (Gonçalves, Souza, & Fernandes, 2022; Mahbod, et al., 2020). Finally, the

computational efficiency of the optimized CNN model highlights its potential for use in low-resource or mobile environments (Li, Yang, feng, Chakradhar, & Zhou, 2016; Rizvi, Rahman, Sheikh, Fuad, & Shehzad, 2023) .

Table 3. Training and Validation Loss Curve

Feature-Engineered Model	Epoch of Convergence	Converge Validation Loss	Visual Behavior
Model 5 (Sigmoid, LR=0.001, Epochs=150)	51	0.07884	Shows a smooth decline in loss; a small spike in validation loss near convergence
Model 11 (Tanh, LR=0.001, Epochs=100)	55	0.8553	Shows early overfitting with a spike at convergence; gap between training and validation loss widens
Model 12 (Tanh, LR=0.0001, Epochs=100)	77	0.5063	Smooth convergence with minimal gap between training and validation loss
Model 15 (Tanh, LR=0.0001, Epochs=150)	43	0.3150	Rapid and consistent convergence; training and validation curves closely aligned

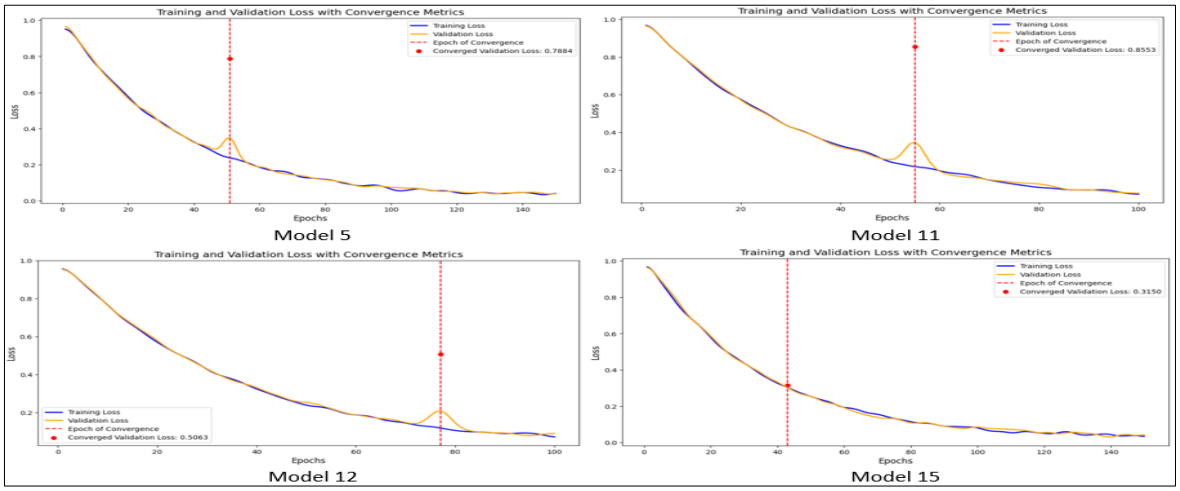

Figure 9. Training and Validation Loss with Convergence of Four (4) Performing Feature-Engineered Models

5. Conclusion

This study developed and optimized a convolutional neural network (CNN) model for classifying Tikog leaves utilized in the banig weaving industry, employing Lab color space, autoencoders, mean-max pooling, and dropout regularization. Among the 27 CNN models tested, Case 12 exhibited the highest performance, demonstrating strong generalizability without overfitting. Notably, this research represents a pioneering effort in applying artificial intelligence to indigenous crafts, fostering technological innovation in cultural preservation. The resulting model shows potential as a practical quality control tool for weavers in Basey, Samar, and similar communities, contributing to sustainable livelihoods and enhancing recognition of native craftsmanship through intelligent systems.

While the study successfully developed and optimized a feature-engineered CNN model for classifying Tikog leaves, several limitations must be acknowledged. First, the dataset was limited to 1,000 raw images (500 per class), which, although augmented to improve generalization, may still constrain the model's ability to accommodate diverse environmental conditions and variations in leaf appearance. Second, the model is specifically tailored to Tikog leaves from the Basey, Samar region, and its performance on similar indigenous materials from other areas remains untested. Third, the model was trained and evaluated in a controlled environment using static images; real-time deployment on mobile or embedded devices with live camera input has not yet been explored. Lastly, while computational efficiency was assessed through floating-point operations per second (FLOPs), other deployment constraints such as energy consumption, latency on edge devices, and hardware compatibility were beyond the scope of this study. These limitations present opportunities for future research, particularly in expanding the dataset, validating the model across different regions, and testing deployment scenarios for mobile or field-based applications.

References

[1] Al-Sabaawi, Aiman, Hassan M. Ibrahim, Zinah Mohsin Arkah, Muthana Al-Amidie, and Laith Alzubaidi. "Amended convolutional neural network with global average pooling for image classification." In International conference on intelligent systems design and applications, Cham: Springer International Publishing, 2020, 171-180.

[2] Arunachalam, A. R. "A survey on deep learning feature extraction techniques." In AIP Conference Proceedings, vol. 2282, no. 1. AIP Publishing, 2020.

[3] Banig. (2013). Retrieved from Explore Basey, Samar Philippines Website: https://evsukraymarabut.wordpress.com/the-famous-banig-in-baseysamar/

[4] Berahmand, Kamal, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. "Autoencoders and their applications in machine learning: a survey." Artificial Intelligence Review 57, no. 2 (2024): 28.

[5] Bharadiya, J. "Convolutional neural networks for image classification." International Journal of Innovative Science and Research Technology 8, no. 5 (2023): 673-677.

[6] Burroughs, Sonya J., Balakrishna Gokaraju, Kaushik Roy, and Luu Khoa. "Deepfakes detection in videos using feature engineering techniques in deep learning convolution neural network frameworks." In 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, 2020, 1-4.

[7] Chatterjee, Rajesh Kumar, Md Amir Khusru Akhtar, and Dinesh K. Pradhan. "Classification and Identification of Objects in Images Using CNN." In International Conference on Artificial Intelligence and Data Science, pp. 16-26. Cham: Springer Nature Switzerland, 2021.

[8] Chen, Shuangshuang, and Wei Guo. "Auto-encoders in deep learning—a review with new perspectives." Mathematics 11, no. 8 (2023): 1777.

[9] Chen, Zhi, Jiang Duan, Li Kang, and Guoping Qiu. "Class-imbalanced deep learning via a class-balanced ensemble." IEEE transactions on neural networks and learning systems 33, no. 10 (2021): 5626-5640.

[10] Duboue, Pablo. The art of feature engineering: essentials for machine learning. Cambridge University Press, 2020.

[11] Fayyad, Usama M., Gregory Piatetsky-Shapiro, and Padhraic Smyth. "Knowledge Discovery and Data Mining: Towards a Unifying Framework." In KDD, vol. 96, 1996, 82-88.

[12] Gholamalinezhad, Hossein, and Hossein Khosravi. "Pooling methods in deep neural networks, a review." arXiv preprint arXiv:2009.07485 (2020).

[13] Gonçalves, Caroline Barcelos, Jefferson R. Souza, and Henrique Fernandes. "CNN architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images." Computers in Biology and Medicine 142 (2022): 105205.

[14] Gulzar, Yonis. "Fruit image classification model based on MobileNetV2 with deep transfer learning technique." Sustainability 15, no. 3 (2023): 1906.

[15] Gulzar, Yonis. "Enhancing soybean classification with modified inception model: A transfer learning approach." Emirates Journal of Food & Agriculture (EJFA) 36, no. 1 (2024).

[16] Gunarathna, M., and R. M. K. T. Rathmayaa. "A review on feature extraction techniques for plant disease classification." In Proc. ICACT, 2020, 118-120.

[17] Gupta, Kavita. "Systematic Review of Contemporary Breast Cancer Detection Techniques Using Machine Learning." In 2024 7th International Conference on Contemporary Computing and Informatics (IC3I), vol. 7, IEEE, 2024, 338-344.

[18] Kishore, Jaydeep, and Snehasis Mukherjee. "Impact of autotuned fully connected layers on performance of self-supervised models for image classification." Machine Intelligence Research 21, no. 6 (2024): 1201-1213.

[19] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86, no. 11 (2002): 2278-2324.

[20] Li, Chao, Yi Yang, Min Feng, Srimat Chakradhar, and Huiyang Zhou. "Optimizing memory efficiency for deep convolutional neural networks on GPUs." In SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2016, 633-644.

[21] Lim, Hyun-il. "A study on dropout techniques to reduce overfitting in deep neural networks." In Advanced Multimedia and Ubiquitous Engineering: MUE-FutureTech 2020, Springer Singapore, 2021, 133-139.

[22] Limonova, Elena, Alexander Sheshkus, and Dmitry Nikolaev. "Computational optimization of convolutional neural networks using separated filters architecture." International Journal of Applied Engineering Research 11, no. 11 (2016): 7491-7494.

[23] Liu, Zhuang, Zhiqiu Xu, Joseph Jin, Zhiqiang Shen, and Trevor Darrell. "Dropout reduces underfitting." In International Conference on Machine Learning, PMLR, 2023, 22233-22248.

[24] Mahbod, Amirreza, Gerald Schaefer, Chunliang Wang, Georg Dorffner, Rupert Ecker, and Isabella Ellinger. "Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification." Computer methods and programs in biomedicine 193 (2020): 105475.

[25] Manaswi, Navin Kumar, and Navin Kumar Manaswi. "Understanding and working with Keras." Deep learning with applications using Python: Chatbots and face, object, and speech recognition with TensorFlow and Keras (2018): 31-43.

[26] Masci, Jonathan, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. "Stacked convolutional auto-encoders for hierarchical feature extraction." In Artificial neural networks and machine learning–ICANN 2011: 21st international conference on artificial neural networks, espoo, Finland, June 14-17, 2011, proceedings, part i 21, Springer Berlin Heidelberg, 2011, 52-59.

[27] Tripathi, Milan. "Analysis of convolutional neural network based image classification techniques." Journal of Innovative Image Processing (JIIP) 3, no. 02 (2021): 100-117.

[28] Mumuni, Alhassan, and Fuseini Mumuni. "Data augmentation with automated machine learning: approaches and performance comparison with classical data augmentation methods." Knowledge and Information Systems (2025): 1-51.

[29] NarasingaRao, M. R., V. Venkatesh Prasad, P. Sai Teja, Md Zindavali, and O. Phanindra Reddy. "A survey on prevention of overfitting in convolution neural networks using machine learning techniques." International Journal of Engineering and Technology (UAE) 7, no. 2.32 (2018): 177-180.

[30] Noh, Hyeonwoo, Tackgeun You, Jonghwan Mun, and Bohyung Han. "Regularizing deep neural networks by noise: Its interpretation and optimization." Advances in neural information processing systems 30 (2017).

[31] Özdem, Kevser, Çağın Özkaya, Yılmaz Atay, Emrah Çeltikçi, Alp Börcek, Umut Demirezen, and Şeref Sağıroğlu. "A ga-based cnn model for brain tumor classification." In 2022 7th International Conference on Computer Science and Engineering (UBMK), IEEE, 2022, 418-423.

[32] Patel, Heena, and Kishor P. Upla. "for Hyperspectral Image Classification." In Computer Vision Applications: Third Workshop, WCVA 2018, Held in Conjunction with ICVGIP 2018, Hyderabad, India, December 18, 2018, Revised Selected Papers, vol. 1019, Springer Nature, 2019,115.

[33] Peng, Min, Yunxiang Liu, Asad Khan, Bilal Ahmed, Subrata K. Sarker, Yazeed Yasin Ghadi, Uzair Aslam Bhatti, Muna Al-Razgan, and Yasser A. Ali. "Crop monitoring using remote sensing land use and land change data: Comparative analysis of deep learning methods using pre-trained CNN models." Big Data Research 36 (2024): 100448.

[34] Prasad, Rai Sachindra. "Retracted: Performance Comparison of HSV and L* a* b* Spaces in Thought Form Image Analysis." In 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE), IEEE, 2017, 310-316.

[35] Qin, Jiaohua, Wenyan Pan, Xuyu Xiang, Yun Tan, and Guimin Hou. "A biological image classification method based on improved CNN." Ecological Informatics 58 (2020): 101093.

[36] Rizvi, Shahriyar Masud, Ab Al-Hadi Ab Rahman, Usman Ullah Sheikh, Kazi Ahmed Asif Fuad, and Hafiz Muhammad Faisal Shehzad. "Computation and memory optimized spectral domain convolutional neural network for throughput and energy-efficient inference." Applied Intelligence 53, no. 4 (2023): 4499-4523.

[37] Rochmawanti, Ovy, and Fitri Utaminingrum. "Chest x-ray image to classify lung diseases in different resolution size using densenet-121 architectures." In Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology, 2021, 327-331.

[38] Safhi, Hicham Moad, Bouchra Frikh, and Brahim Ouhbi. "Assessing reliability of big data knowledge discovery process." Procedia computer science 148 (2019): 30-36.

[39] Saini, Manisha, Preeti Gupta, and Harvinder Kaur. "Comprehensive Study of Indexing Techniques Used for Extracting CNN Features." In 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2023, 59-65.

[40] Sangineto, Enver, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. "Self paced deep learning for weakly supervised object detection." IEEE transactions on pattern analysis and machine intelligence 41, no. 3 (2018): 712-725.

[41] Setyawan, Thomas Agung, Setyo Aji Riwinanto, Arif Nursyahid, and Ari Sriyanto Nugroho. "Comparison of hsv and lab color spaces for hydroponic monitoring system." In 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), IEEE, 2018, 347-352.

[42] Sharma, Anu, and Dharmender Kumar. "Hyperparameter Optimization in CNN: A Review." In 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2023., 237-242.

[43] Shirke, V., R. Walika, and L. Tambade. "Drop: a simple way to prevent neural network by overfitting." Int. J. Res. Eng. Sci. Manag 1 (2018): 2581-5782.

[44] Sornam, M., Kavitha Muthusubash, and V. Vanitha. "A survey on image classification and activity recognition using deep convolutional neural network architecture." In 2017 ninth international conference on advanced computing (ICoAC), IEEE, 2017, 121-126.

[45] Tian, Bing, Liang Li, Yansheng Qu, and Li Yan. "Video object detection for tractability with deep learning method." In 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD), IEEE, 2017.

[46] Vietz, Hannes, Tristan Rauch, Andreas Löcklin, Nasser Jazdi, and Michael Weyrich. "A methodology to identify cognition gaps in visual recognition applications based on convolutional neural networks." In 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), IEEE, 2021.

[47] Wojciuk, Mikolaj, Zaneta Swiderska-Chadaj, Krzysztf Siwek, and Arkadiusz Gertych. "The role of hyperparameter optimization in fine-tuning of cnn models." Available at SSRN 4087642 (2022).

[48] Yadav, P., Madur, N., Dongare, Y., Rajopadhye, V., & Salatogi, S. (2022). Review on Case Study of Image Classification. International Journal of Advanced Research in Science, Communication and Technology, 2(6), https://ijarsct.co.in/Paper5094.pdf, 683-687.

[49] Zafar, Afia, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. "A comparison of pooling methods for convolutional neural networks." Applied Sciences 12, no. 17 (2022): 8643.

[50] Zhang, Jianyu, and Léon Bottou. "Fine-tuning with Very Large Dropout." arXiv preprint arXiv:2403.00946 (2024).

[51] Zhang, Minghua, Yunfang Wu, Weikang Li, and Wei Li. "Learning universal sentence representations with mean-max attention autoencoder." arXiv preprint arXiv:1809.06590 (2018).

[52] Zhang, Wei, Chuanhao Li, Gaoliang Peng, Yuanhang Chen, and Zhujun Zhang. "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load." Mechanical systems and signal processing 100 (2018): 439-453.